GitHub-API-Driven Clustering with 5-level Text Mining Validation Pipeline: R based Approach

Master's Thesis submitted

to

Prof. Dr. Wolfgang Härdle

Supervisor: Lukas Borke

Humboldt-Universität zu Berlin

School of Business and Economics

Institute for Statistics and Econometrics

Ladislaus von Bortkiewicz Chair of Statistics

CASE - Center of Applied Statistics and Economics

by

Anastasia Stepanchenko

(562472)

in partial fulfillment of the requirements

for the degree of

Master of Economics and Management Science

Berlin, June 29, 2016

Acknowledgement

I would like to thank Lukas Borke, theoretical initiator and technological driver, who suggested me a lot of valuable ideas, gave many pieces of helpful advice, who was responsible for support of GitHub repositories and RDC server and via the framework of a larger project generously provided me with GitHup-API data of Quantlets and with results of the massive parallelization.

I am also thankful to Research Data Center (RDC) and Collaborative Research Center (CRC) 649 for system resource and servers. They let this work be completed in finite time.

Abstract

We can not imagine the scientific world today without Big Data. Although it offers wide possibilities for finding patterns of a given matter, it also poses a problem to process huge amounts of raw data and statistical challenges of defining the best model, introducing optimal algorithms and proving the consistency of applied methods. Clustering plays a major role in dealing with high-dimensional data. The aim of this research is to implement initial processing of a massive text sample, to present some results of text mining, to consider the LSA model of dimensionality reduction more precisely, to try different clustering and validation methods in R software environment and to apply these techniques to a collection of numerical methods, called Quantlets.

Keywords: big data, text mining, clustering, clustering validation, high dimensionality, dimensionality reduction, LSA, SVD.

Contents

List of Abbreviations (in order of appearance)

Adj.	Adjusted
API	Application Programming Interface
appr.	approximately
BVSM	Basic VSM
C3	Comfortable, Customizable, Controllable
D3	Data-Driven Documents
DOM	Document Object Model
EI	easy to interpret
et al.	et alia
GVSM	Generalized VSM
HCA	Hierarchical Cluster Analysis
HI	hard to interpret
LSA	Latent Semantic Analysis
QL	Quantlet
QN	Quantnet
SVD	Singular Value Decomposition
TT	Term-term correlations model
Vali-PP	Validation Pipeline
VSM	Vector Space Model
wrt.	with respect to
YAML	YAML Ain't Markup Language

List of Figures

List of Tables

1 Introduction

Today with modern technologies coming into power we can not imagine the scientific world without Big Data. Although it offers wide possibilities for finding patterns of a given matter, it also poses both technical and statistical problems. From a computational point on view, a task could be to process huge amounts of raw data, to minimize storage costs and to make decisions about permissible errors. Statistical challenges, however, are defining the best model, inventing new optimal algorithms and proving the consistency of applied methods. Moreover, one should always be aware of what the final goal of the research is, how (if at all) the achieved results might be helpful for further investigations and whether some useful applications of them could be found.

1.1 Big Project

This research is a part of a bigger project, managed and executed by Lukas Borke, its structure and objectives can be easily described with the diagram below:

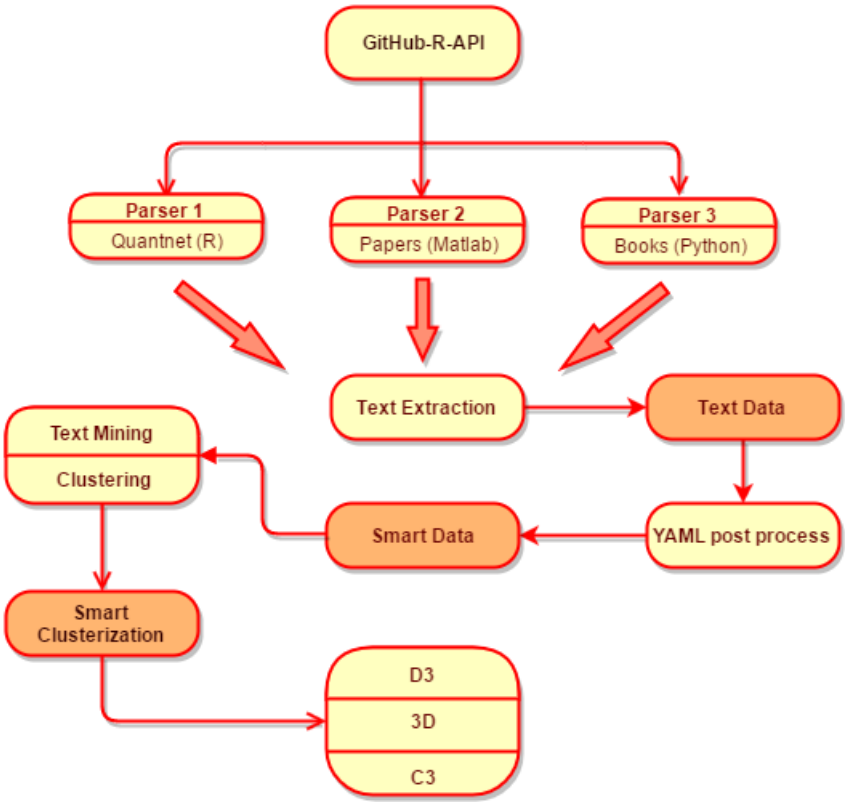


Figure 1: Big Project

The starting point of the big project is **GitHub** - a Git repository hosting service founded in 2008, which not only provides its users with a web-based graphical interface of a well-known version control system, but also allows such high-level features as access control and collaborative work. The first step of the large scheme is to adjust the application program interface (**API**) of GitHub to R software environment and to implement different parsers for extraction of data from various repositories of programming projects. Big Data obtained at this stage is thus constituted by a large corpus of text documents, each of which corresponds to the meta-information of a particular program code.

The next step now is to get Smart Data out of the raw text collection. This task is in our case completed by means of a data serialization language called **YAML**. Designed in 2001 as a human-readable and data-oriented, this language can easily be applied to widely used data frames such as lists and arrays. What makes YAML also rather user-friendly for maintaining hierarchical data is that it avoids the excessive use of brackets, tags and other enclosures which could make the document structure less comprehensible.

Thus derived Smart Data, passed through a complex chain of processing, data mining and clustering, results in a form that can further be brought to life in 3D visualization via opportunities provided by the **D3** and **C3** JavaScript libraries. D3 stands for Data-Driven Documents, and its concept is to perform efficient manipulation of Document Object Model (**DOM**) based on data. It is web-oriented, very fast and efficiently operates with massive datasets. Beta version of Quantnet (see the link in the Appendix) is a good example of D3 in power. C3 is a D3-based chart library, which makes the integration of diagrams into the application more **C**omfortable, **C**ustomizable and **C**ontrollable as is clear from the abbreviation hidden in its name. More information about this big project can be found in the presentation **L. Borke, W.K. Härdle (2015)**.

1.2 Current research

This research starts with the *Parser 1* node of the diagram above and goes along the path till the end point at *Smart Clusterization* node. The aim of it on the first stage is to implement initial processing of a massive text sample obtained from a GitHub mirror of Quantnet(see the link in the Appendix), the later being a collection of numerical methods called Quantlets, and then to illustrate some data statistics.

The second stage is based on clustering, which plays a major role in dealing with high-dimensional data. The idea of combining similar objects into homogeneous groups originated

at dawn of humanity. Being first used for simple classifications, clustering ended up as an integral part of totally different disciplines, among them are archeology, linguistics, bioinformatics, genetics and others (see **B. Everitt, S. Landau, M Leese, D. Stahl (2011)** for more information). Nowadays a vast amount of various numerical methods are introduced in order to make the retrieval of information from the partition easier and more efficient and also to assess *how* efficient it is. Our main goal here was to try different clustering and validation methods in R software environment and to define the most optimal combination of a data configuration, a vector space model, a clustering method and an appropriate way to validate the quality of the resulted partition and then eventually define the reasonable number of clusters. These five elements are combined in what we will call the 5-level Validation Pipeline (or Vali-PP) and which will be further discussed in detail in a separate chapter.

As a result we managed to ground our a priori hypothesis, that LSA model and hierarchical clustering generally perform better for the kind of data considered in this study.

1.3 Outline of the paper

The paper is organized as follows. The next chapter [2] describes main **Theory** behind the research. Section **Data** [3] describes how the data set was obtained and presents some statistics of the text corpus at hand. Chapters **Research** [4] and **Results** [5] describe each step of the investigation and all the outcomes accordingly. Finally, the last section [6] concludes.

2 Theory

2.1 Basic theory

In this chapter we will briefly describe basic theory.

Suppose we have Q – a set of documents (a collection of texts in our case) and T – a set of all the terms in the whole corpus. Let us define $tf(d,t)$ – the absolute frequency of a term $t \in T$ in a document $d \in Q$, i.e. the number of occurrences of this term in a given text document. Those who are familiar with linear algebra may notice, that tf is a homomorphism between the space of word strings and the set of non-negative numbers with respect to concatenation and addition operations accordingly, since:

$$tf(d, t_1 \circ t_2) = tf(d, t_1) + tf(d, t_2).$$

A natural approach would be to represent each text as a vector in a vector space with the number of dimensions equaled to the number of elements in T. Each vector component is associated then with a particular term and its entry is the frequency of this term in a given document. The corpus can now be represented as a matrix D with columns - vectors of the documents. We will call this matrix **"term by document"** matrix, since each column corresponds here to a particular document and each row - to a particular term. As will readily be observed, in most cases such "text" vectors have relatively few non-zero components, which means that the matrix is very sparse.

Another important notion that we use is a **similarity measure**. It is a function defined on a pair of documents and in general form can be written as:

$$S(d_1, d_2) = (Pd_1)^T(Pd_2) = d_1^T P^T P d_2,$$

Each matrix P here defines some so called Vector Space Model (VSM):

$$M_s^P = D^T(P^T P)D,$$

where M_S is a **similarity matrix**.

Those interested in algebraic interpretation can view the original vector space as a space with non-Euclidean metric constructed out of the similarity measure in the following way:

$$d_{orig}(d_1, d_2) = \sqrt{S(d_1 - d_2, d_1 - d_2)} = \sqrt{(d_1 - d_2)^T P^T P (d_1 - d_2)}$$

and P as a mapping $d \to Pd$ to another vector space which is based not only on word frequencies but also preserves some other semantical structure, such as word cooccurrences

or word importance. Furthermore, in this new space the original metric becomes Euclidean, which was pointed out by **Y. Mao, K. Balasubramanian, and G. Lebanon (2010)**:

$$d_{orig}(d_1, d_2) = \sqrt{(P(d_1 - d_2))^T P(d_1 - d_2)} = \|P(d_1 - d_2)\|_2 = \|Pd_1 - Pd_2\|_2.$$

2.2 Vector space models

Depending on matrix P, we will consider three models, mentioned in **N. Cristianini, J. Shawe-Taylor, H. Lodhi (2002)**:

1. **Basic VSM – BVSM**:
   $P = I$, $M_s^{tf} = D^T D$.

   This model was suggested in 1975 by Salton et al. and assumes no vector space mapping. Although BVSM appeared to show good results, it supposes terms to be completely uncorrelated and cannot operate with documents sharing synonyms, the model treats them as unrelated. Hence it can perform well only if documents have a lot of common terms.

2. **Term-term correlations – GVSM (TT)**:
   $P = D^T$, $M_s^{TT} = D^T(DD^T)D$.

   This model was introduced in 1985 by Wong et al. and is an intuitive way to incorporate semantics by means of statistical information obtained from the text database itself. GVSM considers two terms as related proportional to how often they cooccur in the same documents.

   New metric here is defined by DD^T - the so called **"term by term correlation"** matrix, its pq-entry equals zero if and only if no document shares the p-th and the q-th terms. Contrast to basic model, this one can detect similarity of related texts even if they do not have common terms.

3. **Latent Semantic Analysis – LSA**:
   $P = U_k = I_k U^T$, $M_S^{LSA} = D^T(UI_kU^T)D$.

LSA is an alternative approach to preserve semantical structure in a text corpus. It was proposed in 1988 by Deerwester et al. and is based on a singular value decomposition (SVD) of the "term by document" matrix D: $D = U\Sigma V^T$. U, V here are orthogonal matrices and Σ is a diagonal matrix.

P in this model is set to a projection operator onto the first k dimensions: $P = U_k = I_k U^T$ with I_k having only the first k ones on the main diagonal and zeros elsewhere.

In fact, LSA represents a set of models, each variation depending on a particular k. By default, $k = 83$ for our data. Later we will test several values for the number of dimensions in order to get the required results.

As this model performed better than others in the current research we will devote a separate section to it later.

2.3 Clustering

We will compare three different methods: hierarchical clustering, k-means and k-medoids algorithms (or pam - partitioning around medoids). All of them are rather well-known and often used, that is why only the most essential aspects of each method will be pointed out (for more details see **B. Everitt, S. Landau, M Leese, D. Stahl (2011)**).

k-means and k-medoids are examples of so called *hill-climbing algorithms*, that assume the existence of initial partition with a desired number of clusters and then seek to find the best clustering, iteratively rearranging the original partition until the process converges. k-means algorithm tries to refer an observation to the group with the nearest mean, whereas k-medoids takes medoids (the closest objects in terms of a given dissimilarity measure) instead of means, that is, it operates with real elements of the observation set. Both algorithms depend on the initial partition, but if data is structured well enough, one can expect the final result of the clustering procedure to be almost always the same and thus independent of the first conditions. In the Appendix C one can find some QL links to interesting applications of k-means clustering.

In contrast to the k-means and k-medoids methods, the resulting number of clusters for hierarchical procedure is not defined beforehand. The algorithm is rather a number of successive partitions and terminates after the optimum of a chosen criteria is achieved. The method is called *agglomerative* if on the first stage each observation forms its own cluster and are then united step by step. The *divisive* method, on the opposite, starts with only one cluster, containing all individuals and successively divides them into groups. Within

agglomeration hierarchical clustering we used methods called *average*, *ward.D* and *ward.D2*.

As for the number of clusters, in this research we have 500 documents to work with, which is why the possible number varies from 1 (all texts constitute one big cluster, the case is obviously not interesting) to 500 (each text constitutes its own cluster). However, it does not make sense to consider very large numbers of clusters, since we would rather unite, if possible, more documents in order to find and describe some general similarities. For this reason the upper limit of the clusters number in the current paper is set to 250.

2.4 Useful definitions

Further we will use basic definitions, taken in **B. Desgraupes (2013)**.

Suppose we have $\mathbf{N}$ observations of $\mathbf{p}$ variables, divided into $\mathbf{K}$ clusters, $\mathbf{n_k}$ elements in the k-th cluster, $\mathbf{G^{\{k\}}}$ being its barycenter. In geometrical interpretation, we can associate each observation vector x_i and its coefficients with a point $\mathbf{M_i}$ and its coordinates in $\mathbb{R}^p$. Then the coordinates of the barycenter are simply the coefficients of a vector $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$. Denote by $\mathbf{N_W}$ the number of pairs of different points in the set of all the clusters $\mathcal{C}$ and $\mathbf{N_B}$ – the number of pairs of points which do not belong to the same cluster. Let $\mathbf{S_W}$ be the sum of the $\mathbf{N_W}$ distances between all the pairs of points inside each cluster, $\mathbf{S_{min}}$ – the sum of the $\mathbf{N_W}$ smallest distances between all the pairs of points in the full data set and $\mathbf{S_{max}}$ – the sum of the $\mathbf{N_W}$ largest distances between all the pairs of points in the full data set.

To characterize the quality of clustering the following notations are useful. By $\mathbf{T}$ we will denote the total scatter matrix, which equals $\mathbf{N}$ times the covariance matrix of the entire data set. Let **WG** be the within-group scatter matrix. It is the sum of within-group scatter matrices for each cluster, which are proportional to clusters covariance matrices. Later an upper index $\{\mathbf{k}\}$ will mean that a variable takes for computations only elements of the k-th cluster. Denote by **WGSS** the pooled within-cluster dispersion, i.e. the sum of within-cluster dispersions $\mathbf{WGSS^{\{k\}}}$ for all the clusters:

$$WGSS = \sum_{k=0}^{K} WGSS^{\{k\}},$$

$$WGSS^{\{k\}} = \sum_{i \in I_k} ||M_i^{\{k\}} - G^{\{k\}}||^2.$$

On the other hand, define **BGSS** as the between-group dispersion, which geometrically is the sum of the squared distances between the clusters barycenters and the global barycenter,

weighted by the number of observation in the corresponding cluster:

$$BGSS = \sum_{k=1}^{K} n_k ||G^{\{k\}} - G||^2.$$

If we denote by V_j $(j = 1, ..., p)$ a vector in N-dimensional space with coordinates constituted by the values of the j-th variable for each observation, then the total sum of squares **TSS** is equaled to:

$$TSS = N \sum_{j=1}^{p} Var(Vj).$$

2.5 Validation indices

For validation of clustering methods different measures were introduced. We will consider those of them implemented in R packages *clusterCrit* and *NbClust*.

The optimal number of clusters can be derived by maximizing/minimizing of the index value or of the difference between two successive slopes. The last means that on a plot with index values Q against the number of selected clusters K, the best value for K corresponds to an elbow. Suppose, for example, we need to maximize the difference between two successive slopes. Let us denote $V_i = Q_{i+1} - Q_i$, then K is defined by:

$$K = \arg \max_{K_m < i \leq K_M} (V_i - V_{i-1}).$$

Some of 27 internal indices offered by this package are of the exceptional interest allowing rather clear interpretation. The following list contains main formulas and some additional information about each out of 12 measures that we will consider in this research (**B. Desgraupes (2013), G. Brock, V. Pihur, S. Datta, S. Datta (2011)**):

The Ball-Hall index

The Ball-Hall index is the mean, through all the clusters, of their mean dispersion:

$$C = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|^2.$$

In order to define the optimal number of clusters, one should maximize the difference between two successive slopes. The range is $[0, +\infty)$.

The C index

The C index shows for a given clustering which fraction of maximal possible increase over minimal distances constitutes the increase of within-cluster distances between pairs of points over minimal distances. That is:

$$C = \frac{S_W - S_{min}}{S_{max} - S_{min}}.$$

In order to define the optimal number of clusters, one should minimize this index. The range is $[0, 1]$.

The Calinski-Harabasz index

The Calinski-Harabasz index is proportional to the quotient of the between-group dispersion and pooled within-cluster dispersion:

$$C = \frac{N - K}{K - 1} \frac{BGSS}{WGSS}.$$

In order to define the optimal number of clusters, one should maximize this index. The range is $[0, +\infty)$.

The Davies-Bouldin index

The Davies-Bouldin index deals with those clusters which are close in terms of their barycenters to each other but have very distant points within:

$$C = \frac{1}{K} \sum_{k=1}^{K} \max_{k' \neq k} \left(\frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right),$$

where

$$\delta_k = \frac{1}{n_k} \sum_{i \in I_k} \| M_i^{\{k\}} - G^{\{k\}} \|,$$

$$\Delta_{kk'} = \| G^{\{k'\}} - G^{\{k\}} \|.$$

In order to define the optimal number of clusters, one should minimize this index. The range is $[0, +\infty)$.

The Dunn Index

The Dunn Index deals with those clusters which contain the closest points belonging to different clusters and also with clusters that have very distant points within. This index equals the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance:

$$C = \frac{\min\limits_{C_k, C_l \in \mathcal{C}, C_k \neq C_l} \left(\min\limits_{i \in C_k, j \in C_l} dist(i,j) \right)}{\max\limits_{C_m \in \mathcal{C}} diam(C_m)},$$

where $diam(C_m)$ is defined as the maximum distance between observations in cluster C_m. In order to define the optimal number of clusters, one should maximize this index. The range is $[0, +\infty)$.

The McClain-Rao index

The McClain-Rao index is the quotient of the mean within-cluster and between-cluster distances:

$$C = \frac{\frac{S_W}{N_W}}{\frac{S_B}{N_B}} = \frac{N_B}{N_W} \frac{S_W}{S_B},$$

where S_W is the sum of all the within-cluster distances and S_B – the sum of all the between-cluster distances. In order to define the optimal number of clusters, one should minimize this index. The range is $[0, +\infty)$.

The Ratkowsky-Lance index

The Ratkowsky-Lance index is based on the mean $\overline{R}$ of the quotients between BGSS and TSS for each variable of the data:

$$TSS_j = NVar(Vj), \quad BGSS_j = \sum_{k=1}^{K} n_k (\mu_j^{\{k\}} - \mu_j)^2,$$

$$\overline{R} = \frac{1}{p} \sum_{j=1}^{p} \frac{BGSS_j}{TSS_j}$$

$$C = \sqrt{\frac{\overline{R}}{K}}.$$

In order to define the optimal number of clusters, one should maximize this index. The range is $[0, +\infty)$.

The Ray-Turi index

The Ray-Turi index is a quotient between two quantities: the mean of the squared distances from all the points to the barycenter of their cluster and the minimum of the squared distances between the cluster barycenters:

$$C = \frac{1}{N}\frac{WGSS}{\min\limits_{k<k'}\Delta^2_{kk'}}.$$

In order to define the optimal number of clusters, one should minimize this index. The range is $[0, +\infty)$.

The Silhouette index

The Silhouette index operates with quantities that only depend on the average distances between a given observation and other observations inside its own and also inside the nearest cluster.

$$C = \frac{1}{K}\sum_{k=1}^{K}\mathfrak{s}_k,$$

where

$$\mathfrak{s}_k = \frac{1}{n_k}\sum_{i\in I_k} s(i), \quad s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))},$$

$$a(i) = \frac{1}{n_k - 1}\sum_{i'\in I_k} d(M_i, M_{i'}), \quad b(i) = \min_{k'\neq k}\frac{1}{n_{k'}}\sum_{i'\in I_{k'}} d(M_i, M_{i'}).$$

In order to define the optimal number of clusters, one should maximize this index. The range is $[0, 1]$.

The Trace W index

The Trace W index is simply the pooled within-cluster dispersion:

$$C = \mathrm{Tr}(WG) = WGSS.$$

In order to define the optimal number of clusters, one should maximize the difference between two successive slopes. The range is $[0, +\infty)$.

The Wemmert-Gancarski index

The Wemmert-Gancarski index is based on quotients of distances between the points and the barycenters of all the clusters.

$$C = \frac{1}{N} \sum_{k=1}^{K} \max\{0, n_k - \sum_{i \in I_k} R(M_i)\},$$

where

$$R(M) = \frac{\|M - G^{\{k\}}\|}{\min\limits_{k' \neq k} \|M - G^{\{k'\}}\|}$$

In order to define the optimal number of clusters, one should maximize this index. The range is $[0, 1]$.

The Xie-Beni index

The Xie-Beni index is the quotient between the mean pooled within-cluster dispersion and the minimum of the minimal squared distances between the points in the clusters.

$$C = \frac{1}{N} \frac{WGSS}{\min\limits_{k<k'} \min\limits_{\substack{i \in I_k \\ j \in I_{k'}}} d(M_i, M_j)}$$

In order to define the optimal number of clusters, one should minimize this index. The range is $[0, +\infty)$.

3 Data

The whole process of obtaining data can be divided into the following stages: decision about Big Data, step to Smart Data, text database collection, choice of the sample configuration, text preprocessing and weighting. We will discuss each stage in the following sections.

3.1 Getting data set

Big Data. The initial source of data is a GitHub mirror of Quantnet (later **QN**) - an online repository of numerical methods (we will call them Quantlets, later **QLs**). QLs consist of documents related to various scientific topics, created by different authors from professional researchers to university students. These documents may contain source code created in such software environments as Matlab, R, SAS, Python, Gauss, Stata and XploRe. This research deals with those of them written in R, Matlab, SAS and Python.

Smart Data. For Big Data processing, a new format of QL metadata was introduced. In order to standardize all the documents, we supplied each QL with an individual meta-information text file in a format of YAML serialization language. Each meta-info file contained five *mandatory fields* (see the link to Styleguide for meta-information and an example of a well-made metainfo.txt in the Appendix C):

- "Name of Quantlet",
- "Published in" - the title of the original book or paper,
- "Description" (at least 10 words),
- "Keywords" (at least 5) - words chosen from the global QN keyword list,
- "Author" - chosen from the global QN author list.

Among *optional fields* are:

- "See also" - other QLs with related topics,
- "Datafile" - all data files from the global QN data file repository used in the code,
- "Example" - a list of generated plots and their descriptions.

Text Database. For the research purpose 500 standardized QLs were collected via a GitHub-API Parser implemented in R. Although we only worked with the fields mentioned above, the parser included some other technical information into the data set. Both types of the fields were exposed to the initial preprocessing and deriving some basic statistics.

Text preprocessing. In order to make texts comparable and hence to obtain unbiased results, the corpus was cleaned of the following items: punctuation, white space, numbers, capital letters (changed to lowercase), prefixes and endings (stemming based on Porter's algorithm) and stopwords (i.e. words that are commonly used and cannot characterize text properly).

Weighting. A term-document matrix was weighted by the product of the following specified components:

- a weighting schema for **term frequencies** was set to **natural**,
- a weighting schema of **document frequencies** was set to **identity**,
- a schema for **normalization** was set to **cosine**.

3.2 Data statistics

At the initial preprocessing stage we wanted to identify fields that were equivalent although formally written in different ways. The bar chart on **Figure 2** below shows how many documents have a given field as it was in the beginning.

From such a distribution and names on the horizontal axis, it was clear that some fields had to be merged. The resulting field received the name which corresponded to the maximal number of documents including it, namely:

- "Author [New]", "Author[Update]" became just "Author",
- "Datafiles", "Datafile[example]" became "Datafile",
- "Keywords[new]" became "Keywords",
- "Name of QuantLet" became "Name of Quantlet",
- "Subfunctions" became "Subfunction".

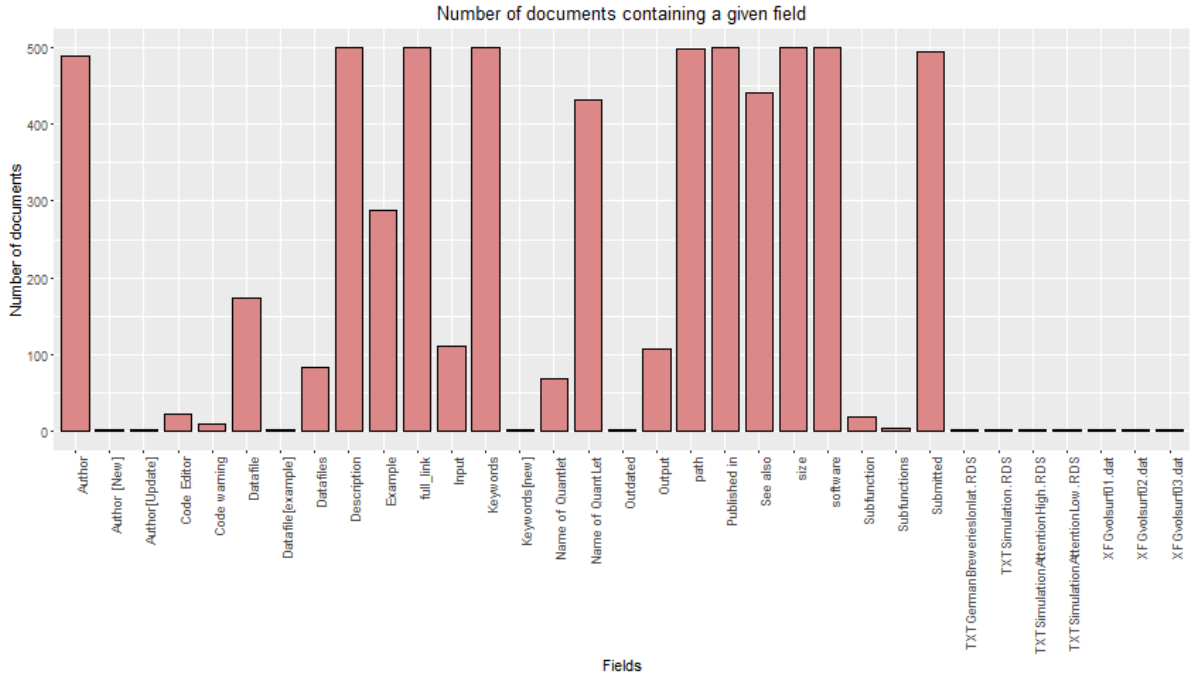


Figure 2: Number of documents containing a given field

The result can be seen on the following diagram sorted in alphabetical order:

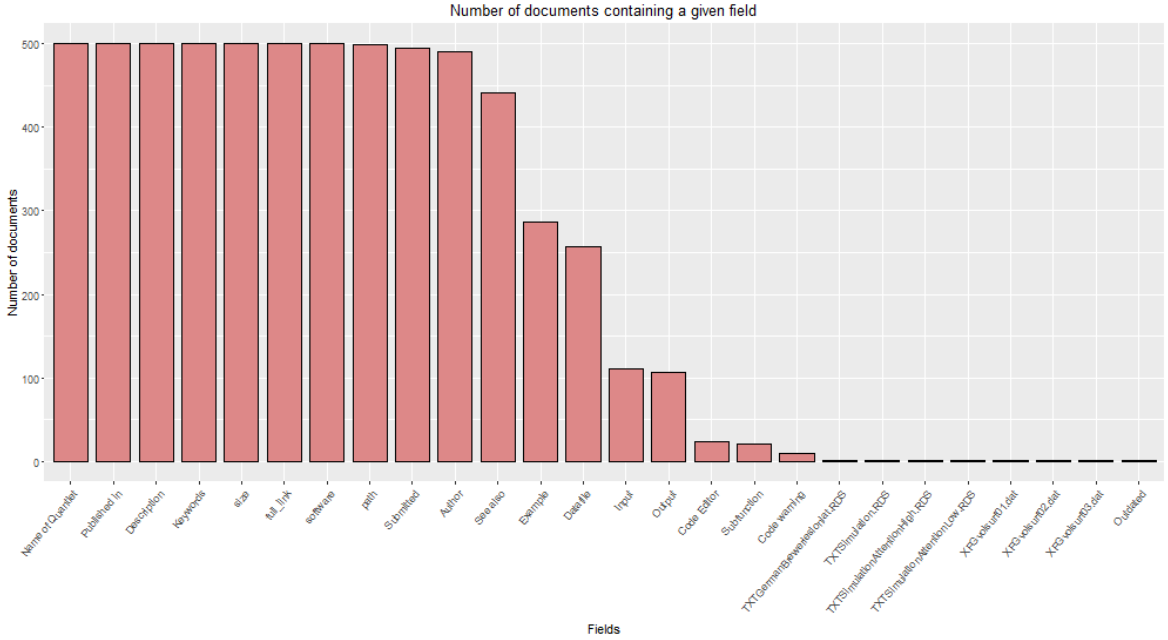


Figure 3: Number of documents containing a given field (merged)

Another idea which we followed was that though some fields had been formally included into meta-information files, they could actually appear to be empty over the negligence of the person who submitted the Quantlet. On the next step such categories were excluded and the final state of the data set is shown on the **Figure 4** below. One can see that now there are three fields less on the axis.

On the next step we wanted to count how many char symbols each field embraces in the whole data set since it was useful for making decision about the weighted configuration for the further computations. From the **Figure 5**, which presents the required histogram, one can derive, that the 5 most significant for clustering fields could be: "description", "keywords", "see also", "example", "datafile" and "author".

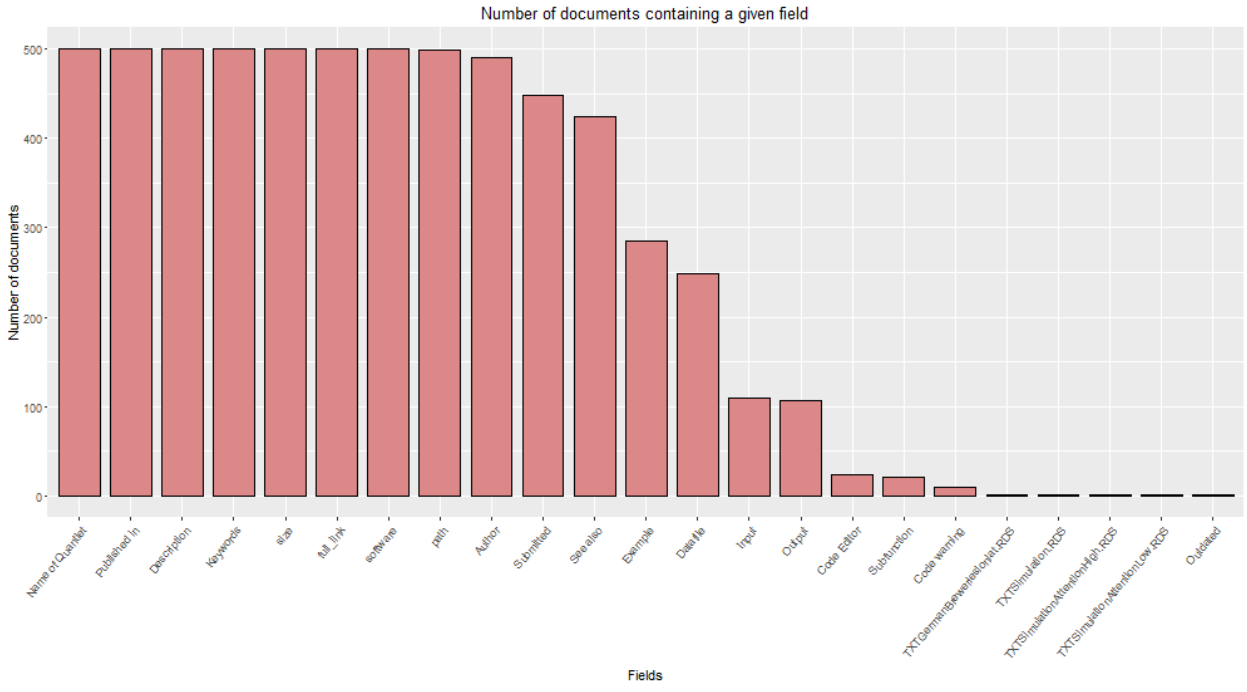


Figure 4: Number of documents containing a given field (cleaned and merged)

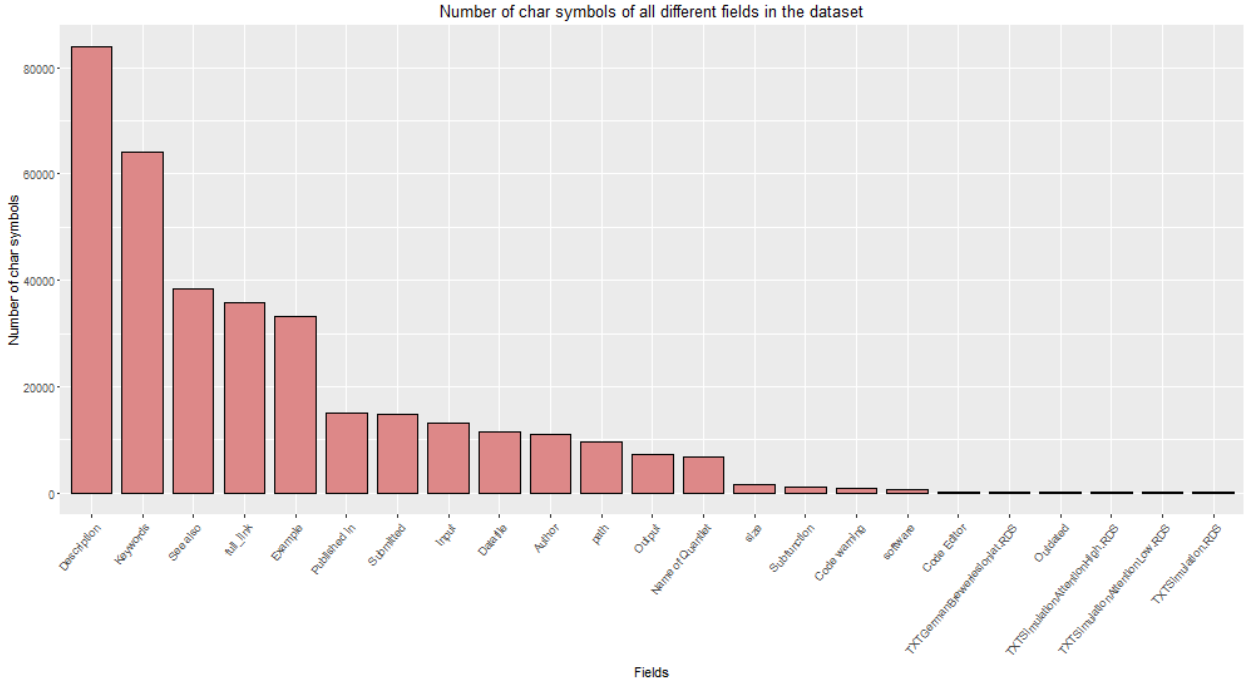


Figure 5: Number of char symbols of all different fields

3.3 Sample configuration

By configuration we mean a combination of fields which are left out of the whole meta-information and only this combination now stands in for each document. In frames of our study we defined three sample configurations, two extreme cases and one of our own choice:

1. **Configuration I** Text documents are represented by descriptions and keywords only, this case is thus minimum that could be reasonable for text mining.

2. **Configuration II** Text documents are represented by all (not technical) fields of meta-info, and this case is obviously the maximal one, including though a considerable amount of unimportant information.

3. **Configuration III** Text documents are represented by weighted combinations of the most significant fields, determined in the previous section. Weights chosen were (6, 10, 3, 4, 5, 4) for the fields: "description", "keywords", "see also", "example", "datafile" and "author" correspondingly. The logic behind such a choice is that description and especially keywords reflect the most profound essence of the text and must thus make the

biggest contribution. The next most important category is "datafile", since several QLs related to the same observation set should be more or less close to each other. Slightly less significant are "author" and "example", because the author can also submit QLs from a completely different area and "example" often just copies the "description". Then at the end comes "see also" with only names of the similar documents in it.

4 Research

4.1 Validation Pipeline

To summarize what has already been said, the research deals with the 5 following features:

- 3 **Sample configurations** formed in chapter **Data**;
- 4 **Vector space models** determined in chapter **Theory**;
  - BVSM,
  - GVSM (TT),
  - LSA (automatic choice of the dimension),
  - LSA (own choice of the dimension);
- 12 **Indices** to validate the quality of clustering (defined in chapter **Theory**);
- 3 different **Clustering methods**:
  - Hierarchical Clustering (average, ward.D, ward.D2 algorithms),
  - The k-Means Clustering,
  - The k-Medoids Clustering;
- **Number of clusters**: from 2 to 250.

The main goal of this work was to find such a combination of all 5 components listed above, that would give the best clustering, when applied to smart data. Visual interpretation of this idea could be a pipe with 5 gear wheels (each of them representing one component), that takes smart data as an input and returns smart clusterization as an output. Let us call this newly invented device the **Validation Pipeline** (Vali-PP).

The main purpose in this analogy is to determine the best angle of rotation for each gear wheel, so that together their combination lets smart data pass through the pipe in the most effective way. Effectiveness here means that we should try to find the maximal amount of information and validation techniques that can be omitted because of storage and computational costs, but without changing the global results and conclusions.

A nice illustration of the described process can be found below.

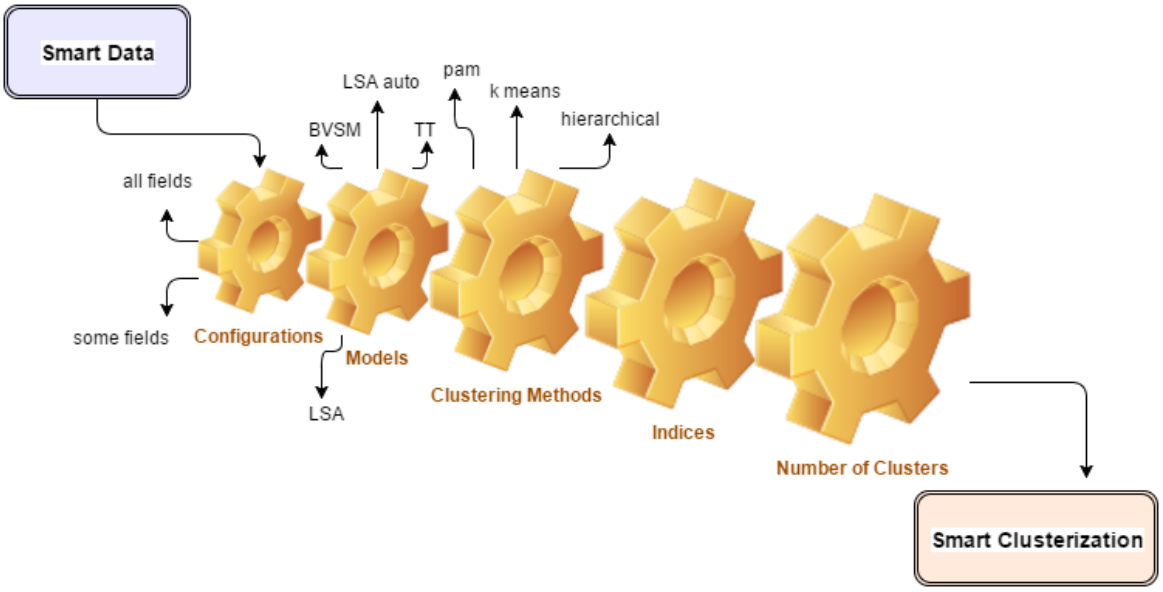


Figure 6: Vali-PP

4.2 LSA anatomy

Before starting the clustering validation part of our research, we would like to devote a section to the LSA model, since the initial hypothesis was that it should theoretically perform better and which eventually turned out to be true.

Just to remind, the main idea in this model is to reduce dimension by leaving only the k largest (in terms of the absolute value) singular values from the diagonal matrix in the SVD decomposition (for a mathematical definition of LSA see chapter **Theory** [2]). Therefore the first thing that comes to mind is the effect of singular values and the way to represent them for the clear understanding of how many dimensions should be kept in the resulting space. On the following **Figure 7** the weights of the singular values for our data are plotted. From this can be derived that the first six elements are particularly large and all other singular values starting from the seventh one form an almost continuous curve.

But to leave such a small number of dimensions would lead to a great loss of data information, which is why LSA is always a compromise and the reduction error should also be taken into consideration. Automatic mode of the *lsa* procedure in R uses a *dimcalc_share*() function to determine an optimal number of clusters. This method takes all the singular values sorted in the descending order, calculates step by step the sum of the first elements and terminates when this sum constitutes the given share of the total singular values sum. This share equals 0.5 by default and in our case results in 83 dimensions. **Figure 8** illustrates

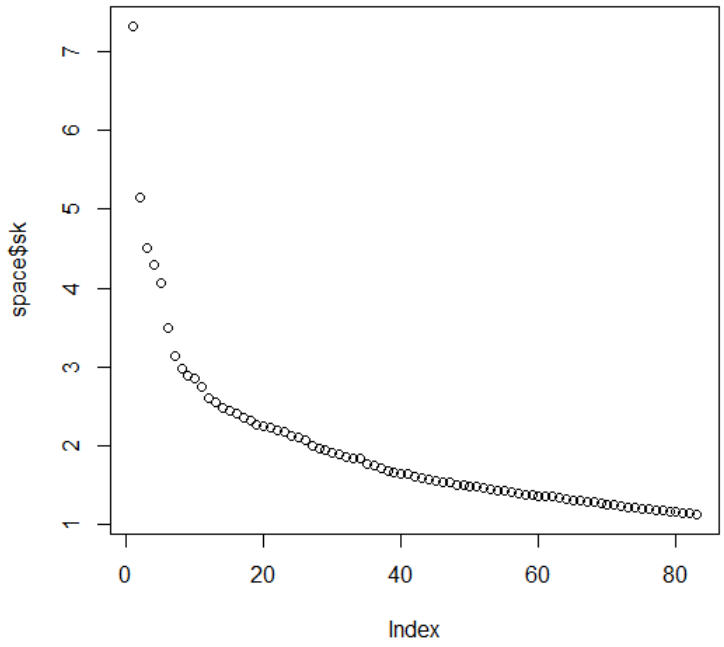
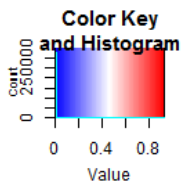
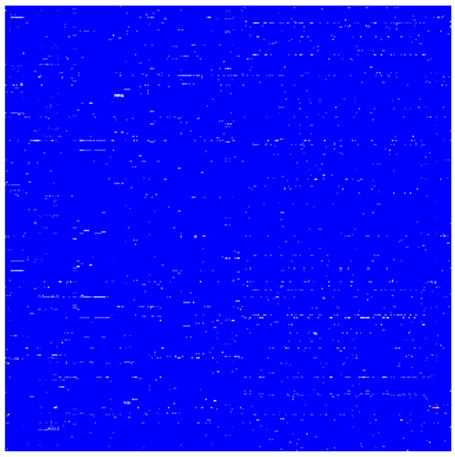
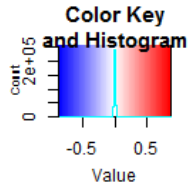


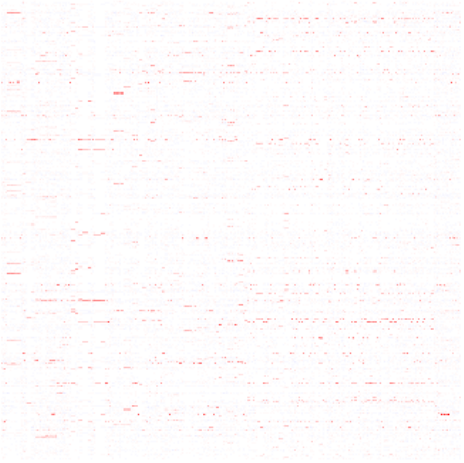
Figure 7: Weights of singular values

a distribution of the matrices in the following order: the original term-document matrix, the truncated one from the automatically created LSA space and the difference matrix for the first two matrices.









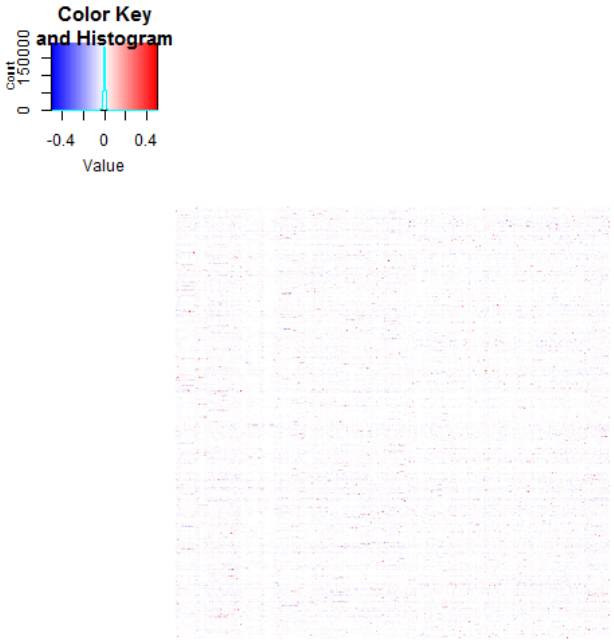


Figure 8: The error of approximation in matrix form

It could also be interesting to make some visualizations of the LSA components themselves. Shown on the **Figure 9** below are the distribution heatmaps (in the corresponded order): of the original term-document matrix, of the first orthogonal matrix U, whose columns represent the "semantic components", of the diagonal matrix Σ with descending singular values and of transposed orthogonal matrix V whose rows represent the "document coefficients" if we regard the product $U\Sigma$ as a new basis of the LSA space, formed by these semantic components.

Although the distribution of all matrix entries is very close to normal, some "structures" are recognizable. For example, left columns (i.e. columns corresponded to the larger singular values) show clearer concentration on several terms. The more it tends to the right, the more diffuse and chaotic becomes the behavior of term frequencies. An intuitive explanation could be that the right components (those with less weight of singular values) contribute consistently, but this contribution is small. Besides that, a lot of term values of the left components (which are actually represented by row coordinates of the matrix) are close to zero. We can thus conclude that left components influence some terms especially strong through all the documents.

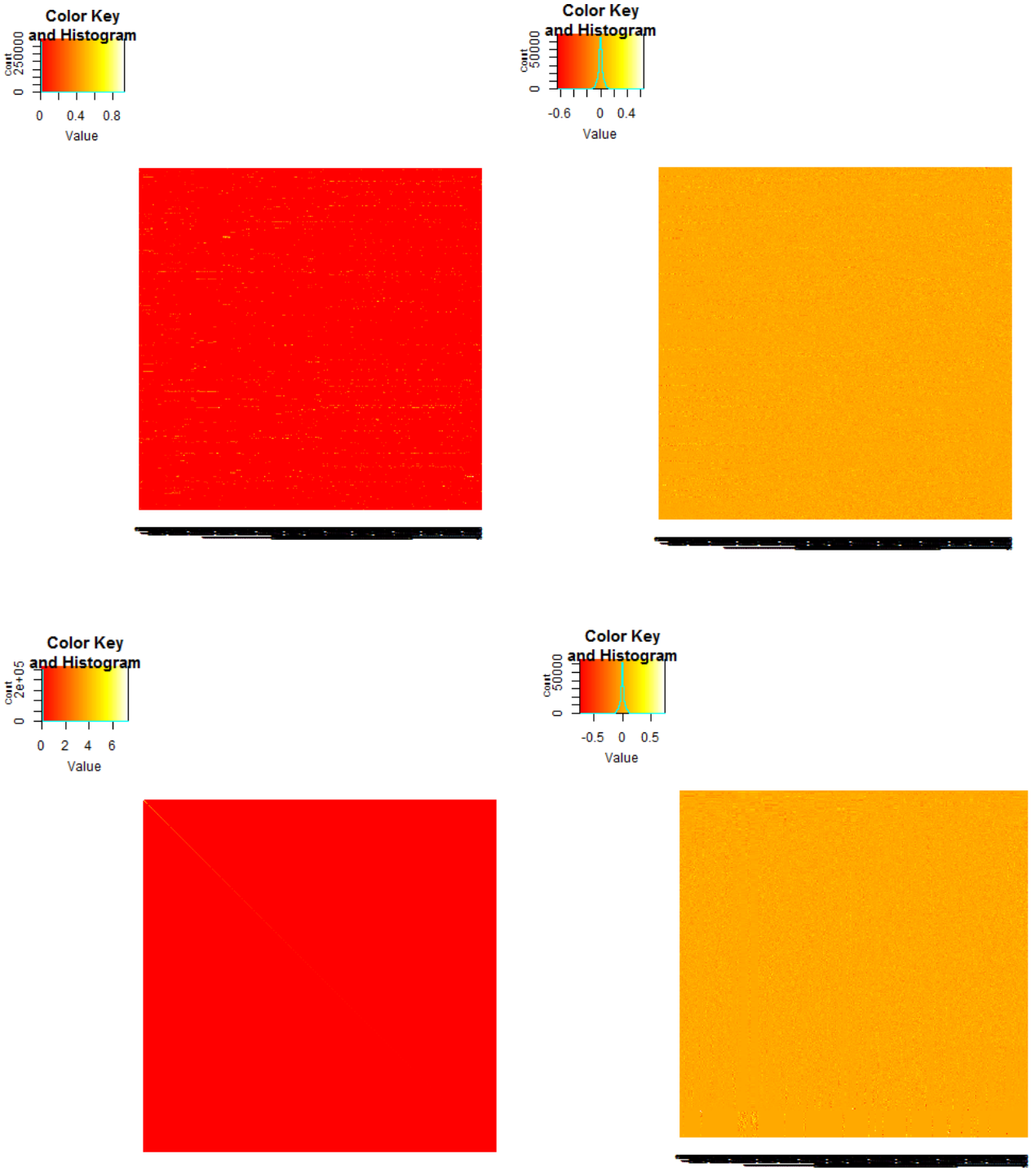


Figure 9: The distribution of SVD

On the picture related to the diagonal matrix a slightly noticeable change of color can be seen in the top left area, which corresponds to the most significant singular values.

As for the matrix V^T (the right matrix in SVD), the distribution of all entries is also very

close to normal, but no "structures" are identifiable. For this reason the entropy of each row is the same and so is the "document entropy" of each individual semantical component. Since the distribution of each row is even close to normal distribution, maximal entropy could be reached in each of them.

4.3 Clustering validation

For the clustering validation part of this research, two R packages were used: *clusterCrit* and *NbClust*. With the first library we calculated all the validation indices that were defined in theoretical part of this paper except for Silhouette. From the second package those seven measures were taken, which intersect with measures from *clusterCrit*, namely: Ball-Hall, C-Index, Calinski-Harabasz, McClain-Rao, Ratkowsky-Lance, Davies-Bouldin, Dunn and also Silhouette. We had three reasons to include another library into our study:

1. to compare results for the same validation methods but obtained from different packages,
2. to get results for Silhouette index which had an error in the *clusterCrit* implementation,
3. to test whether those measures that did not give clear results in *clusterCrit* could be easier to interpret in another package.

In frames of the package *clusterCrit* we compared k-means, k-medoids and *average* hierarchical clustering methods, whereas for *NbClust* methods *ward.D*, *ward.D2* and k-means were selected.

Among other R packages used in calculations were:

- ***tm*** - a text mining library, that provided us with such means as data import, corpus handling, preprocessing and construction of term by document matrices,
- ***SnowballC*** gave an access to Porter's word stemming algorithm and thus an opportunity to operate with different forms of words,
- ***lsa*** created a latent semantic space from a given term by document matrix,
- ***snowfall*** - a package presents methods for development of parallelized processes, includes extended error checks and some other functions,
- ***cluster*** - an auxiliary library for cluster analysis,

- ***abind*** helped to merge multidimensional arrays obtained after the parallelization into a single one,
- ***ggplot2*** produced plots.

Since the calculations were time consuming (in case of *NbClust* it sometimes took more than 4 hours to compute results for a single index, the execution of *clusterCrit* ran slightly faster, but still lasted appr. 10 hours for all the indices), the whole process was parallelized over 24 CPUs and terminated altogether less than in a day.

Further considerations and conclusions were made based on plots with 4 curves (a curve for each vector space model) showing the value of a given validation index (axis Y) for each number of clusters from 2 to 250 (axis X). Such plots were created for each combination of a clustering method and a sample configuration. That is, we had nine pictures for each measure, which gave $9 * 11 = 99$ pictures for the package *clusterCrit* and another $9 * 8 = 72$ for the package *NbClust*. As a result, we dealt with 171 plots. The most demonstrative of them will be presented in the next chapter, for the rest follow the link to the project source in the Appendix C.

5 Results

5.1 Preliminary results

It has already been mentioned that a priori hypothesis implied that the LSA model will in most cases lead to the best clustering. This prediction was based on our preliminary study, which aimed to try some R packages devoted to text mining and clustering and to apply these techniques to W. Shakespeare's collection of works. In frames of that research the text corpus was relatively small, consisting only of 36 poems (that is, almost 14 times less elements than in the current data set), but at the same time each text had strongly pronounced semantical structure, which made interpretation of clusters clearer.

For clustering at that time an R package *clValid* was involved. Vector space models and clustering methods were the same as now, however the library made only three internal measures available: Connectivity, Silhouette Width and Dunn Index.

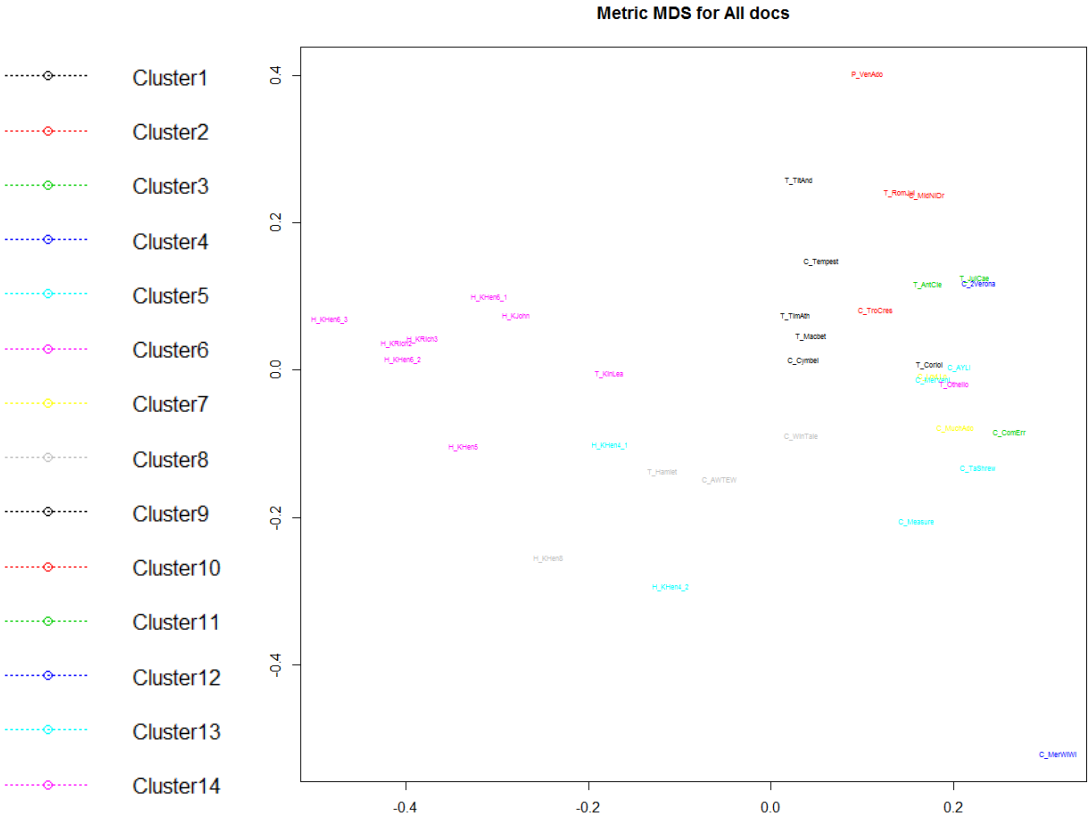


Figure 10: Shakespeare's works MDS plot

The most appropriate combination of a model and a method appeared to be LSA and k-means. For 36 documents it was even possible to present a nice visualization of clustering.

We also find that it is a good opportunity to demonstrate the advantage of D3 charts over the MDS (Multidimentional Scaling) plots, which you can see on the **Figures 10 and 11** correspondingly.

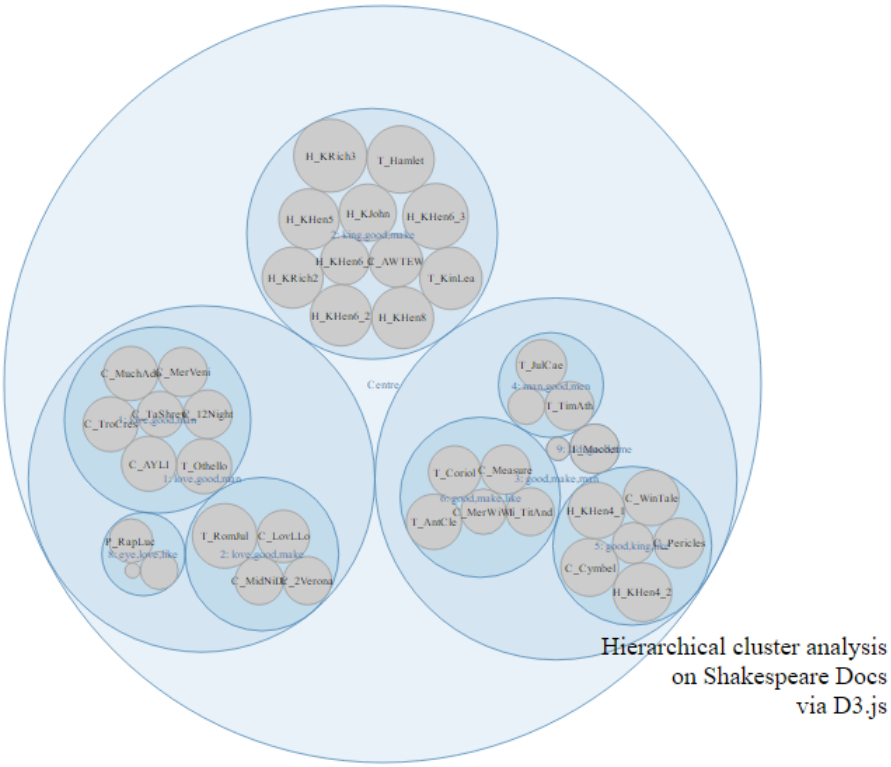


Figure 11: Shakespeare's works D3 chart, taken from the beta version of Quantnet

5.2 Results for *clusterCrit* package

The **Table 1** below shows the best clustering method, the best model and the best configuration for each index. The last two wheels of the Vali-PP (the best index and the most optimal number of clusters) are to be considered in each particular case separately. However, some indices in the *clusterCrit* implementation do not allow making a single decision: Davies-Bouldin, Ray-Turi, Xie-Beni and Dunn. In these cases necessary remarks are provided. **EI** - easy to interpret, **HI** - hard to interpret.

Plots for HI indices looked chaotic and unstable, see **Figure 12** below as an example, showing 9 pictures, corresponded to the Xie-Beni index, rows relate to different sample configurations. Here only hierarchical clustering (the very right column) leaves a chance to determine the best model.

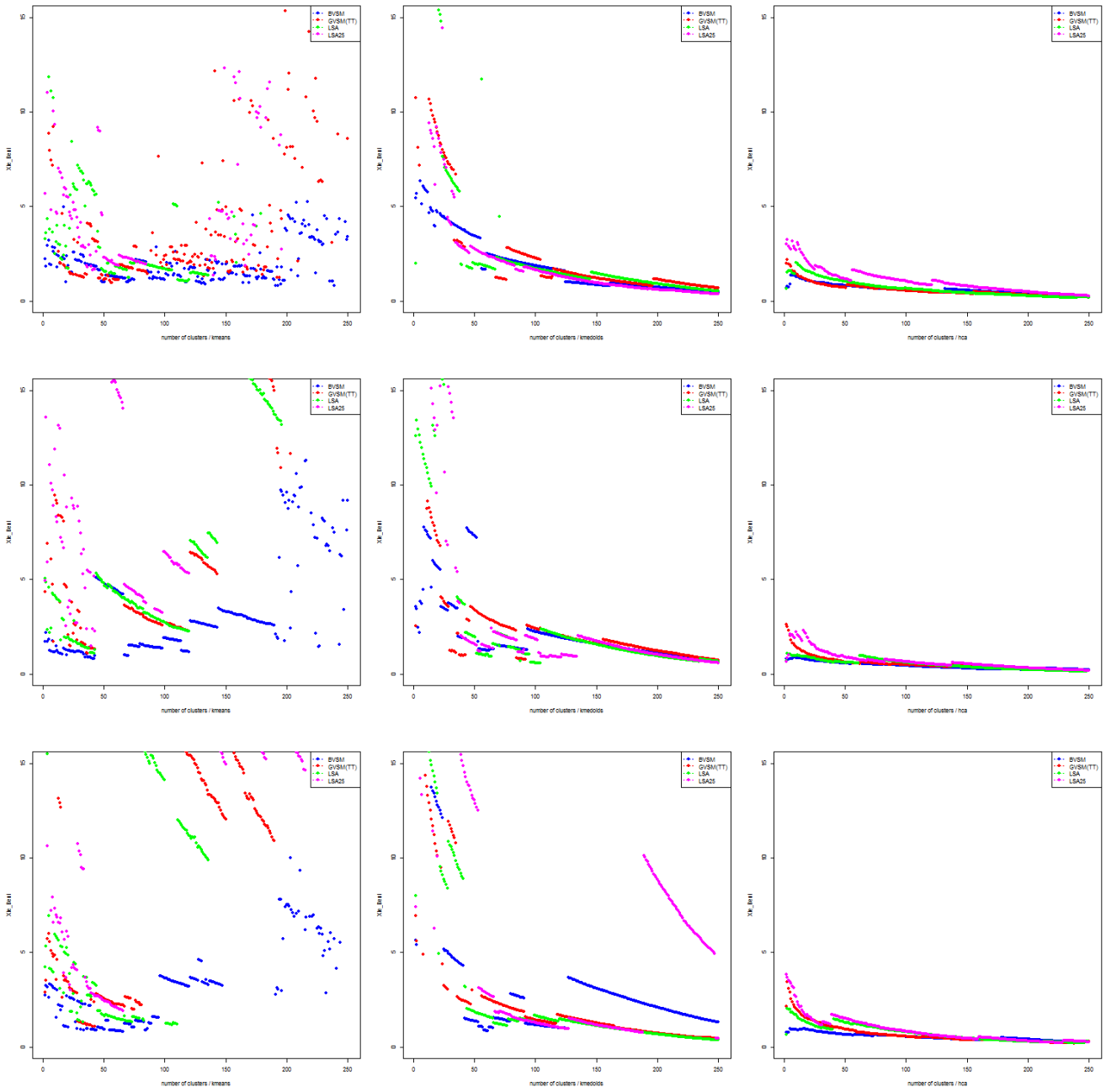


Figure 12: The Xie-Beni index - to minimize

Just to compare how very different look pictures for EI indices, we also wanted to present the same set of plots for the Calinski-Harabasz index. From it, one can easily derive that the best model is LSA with 25 dimensions (later LSA25), the best method is k-medoids, but

hierarchical clustering is comparable from a certain cluster number size and finally the best configuration is the II, since the index is to be maximized and the curves lie higher in the second row.

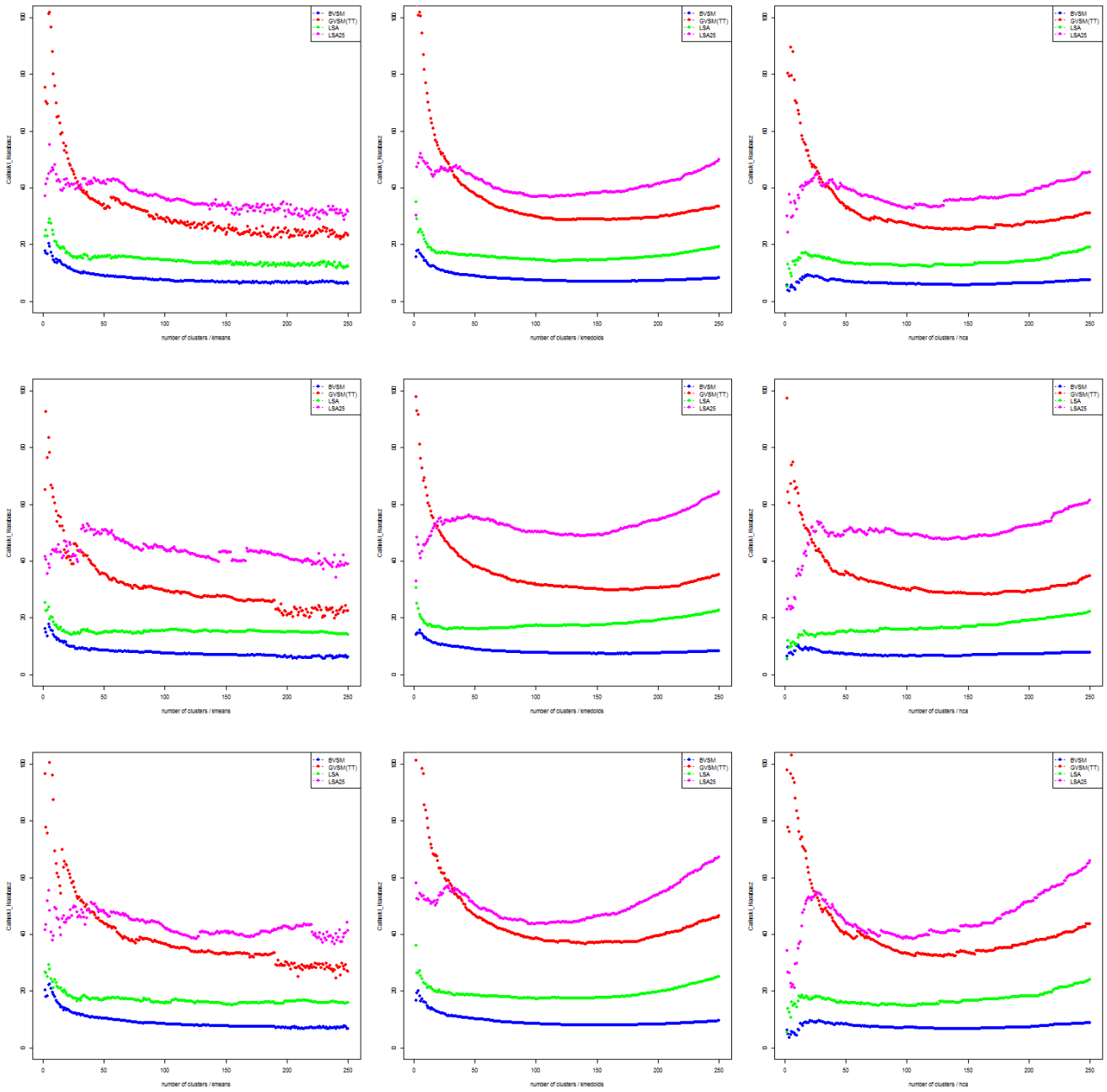


Figure 13: The Calinski-Harabasz index - to maximize

Now we will give some comments to the main results table itself: hca in the third column obviously stands for hierarchical clustering analysis, all the footnotes are discussed later.

Index	Best model	Best clustering method	Best conf.
Ball-Hall (EI)	TT[1]/LSA25[2]	hca	II
C-Index (EI)	TT/LSA25	hca	III
Calinski-Harabasz (EI)	LSA25	k-medoids[3]	II
McClain-Rao (EI)	LSA25	k-means/k-medoids[4]	III
Ratkowsky-Lance (EI)	TT/LSA25	all	I/II
Trace-W (EI)	LSA/LSA25	hca	II/III
Wemmert-Gancarski (EI)	LSA25	k-mean/k-medoids[5]	III
Davies-Bouldin (HI)	LSA25	k-medoids[6]	II
Ray-Turi(HI)	LSA25	hca	III
Xie-Beni (HI)	BVSM[7]	hca	II
Dunn (HI)	BVSM[8]	hca	II

Table 1: Main results table for *clusterCrit*

1. TT model is the best wrt. optimal cluster number selection, whereas
2. LSA25 model is the best wrt. global behavior of the curves;
3. k-medoids method is better, but hca is comparable from a certain cluster number size;
4. hca is also comparable in LSA25, which is the best model for this index;
5. hca is here comparable wrt. global behavior of the curves;
6. hca shows more stable behavior in all the models;
7. all the models are very close in hca, in particular LSA and BVSM;
8. all the models, in particular BVSM and LSA, are relatively close and are also in a quite small interval, regarding that the value range of this index is $[0, +\infty)$.

Overall conclusion, that can be made based on the table above, is that even if the advantage of LSA and hca separately is sometimes not obvious, their combination is never worse than any other combination of a similarity model and a clustering method. As for the sample configuration, it is hard to decide, whether II or III is better, the only thing which becomes clear is that including minimum information for clustering is not a good decision.

5.3 Results for *NbClust* package

Table 2 shows a similar gathering of results for the *NbClust* package. Comments to the footnotes and overall conclusion are almost the same:

Index	Best model	Best clustering method	Best conf.
Ball-Hall (EI)	TT[1]/LSA25[2]	all	I/II/III
C-Index (HI)	LSA25	Ward D	II/III
Calinski-Harabasz (EI)	LSA25	Ward D/ Ward D2	II
McClain-Rao (EI)	LSA25	Ward D2	III
Ratkowsky-Lance (EI)	TT/LSA25	all	I/II/III
Davies-Bouldin (EI)	LSA25	k-means	I/II/III
Silhouette (EI)	LSA25	Ward D/ Ward D2	III
Dunn (HI)	BVSM[3]	Ward D2	II

Table 2: Main results table for *NbClust*

1. TT model is the best wrt. optimal cluster number selection, whereas
2. LSA25 model is the best wrt. global behavior of the curves;
3. all the models, in particular BVSM and LSA, are relatively close and are also in a quite small interval, regarding that the value range of this index is $[0, +\infty)$.

To summarize, we can say that here it is even separately obvious that LSA model and hierarchical clustering analysis are better than other models and clustering methods. Configuration I again appeared to be the worse. An interesting point though is that whether indices are easy or hard to interpret depends not on their definitions but on their implementations.

6 Conclusions

To summarize what has been done in frames of the current study, it can be said that the main goal of the research was achieved: we introduced and developed the so called Validation Pipeline – a functional multi-staged instrument for clustering analysis and determined with its help the most appropriate way to represent data (namely, LSA model together with a sufficient set of fields to reflect the core semantics of a text corpus) and at the same time made a decision about the most optimal clustering algorithm.

Effective applications of our findings already exist and can be shown by the earlier mentioned beta version of Quantnet, which uses smart clustering for D3 visualization of numerical methods. **Figure 14** illustrates how it works.

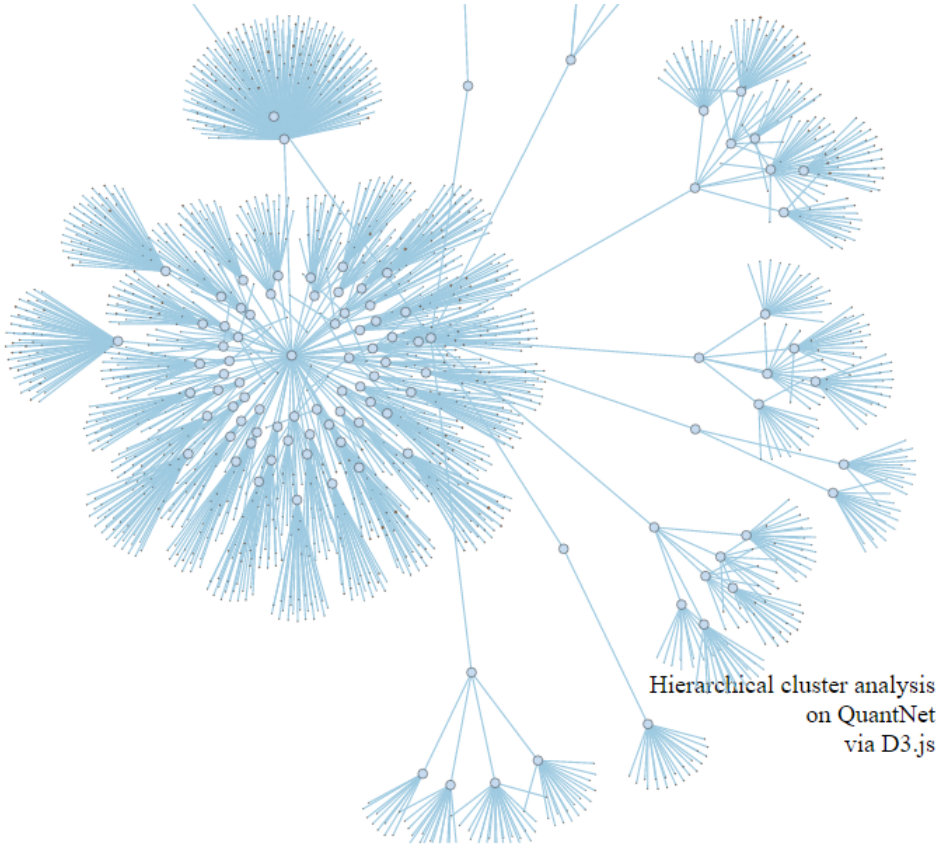


Figure 14: Quantnet D3 chart, taken from the beta version of Quantnet

It is also worth to point out that the whole organization of this project, including version control of all source files, updates of the LaTeX draft for the paper and keeping results

relevant on each stage of the investigation, was itself done using high possibilities provided by GitHub.

Among limitations of this study one can name a relatively narrow variety of texts suitable for the introduced technologies. By now the procedures are driven only by meta-information files of a beforehand specified format. As a consequence, parsers are not universal and have to be adapted for every particular case.

Based on such thoughts, several directions of how to develop the current research further could be issued. One of the possible suggestions is to implement other types of parsers, which would fit more different GitHub organizations and also to extend smart text structures, thus theoretically allowing clustering results better reflect the reality.

Another interesting issue could be to investigate how the error due to dimensionality reduction of the LSA model depends on the number of resulting dimensions. For example, a choice of an appropriate matrix norm might be helpful here for estimation of the error in a matrix form.

In the sphere of experiments with text data it would be useful to test the behavior of results when the set of terms is randomly changed, for example, if we get rid of some words or, on the contrary, add some new terms. This could give the answer to question, whether mistakes caused by negligence are really significant.

References

B. Desgraupes (2013): *Clustering Indices*, University Paris Ouest, Lab Modal'X.

B. Everitt, S. Landau, M Leese, D. Stahl (2011): *Cluster Analysis*, UK: Kings College London, 5th ed.

G. Brock, V. Pihur, S. Datta, S. Datta (2011): *clValid, an R package for cluster validation*, Department of Bioinformatics and Biostatistics, University of Louisville.

L. Borke, W.K. Härdle (2015): *D3-3D-LSA for QuantNet 2.0 and GitHub*, Ladislaus von Bortkiewicz Chair of Statistics, C.A.S.E. Center for Applied Statistics and Economics, Humboldt Universität zu Berlin.

N. Cristianini, J. Shawe-Taylor, H. Lodhi (2002): *Latent Semantic Kernels*, University of London.

Y. Mao, K. Balasubramanian, and G. Lebanon (2010): *Dimensionality Reduction for Text using Domain Knowledge*, Georgia Institute of Technology.

A Example of Metainfo.txt

Name of QuantLet: MVAsirdata

Published in: Applied Multivariate Statistical Analysis

Description: Generates a data set and applies the sliced inverse regression algorithm (SIR) for dimension reduction.

Keywords: EDR-directions, dimension-reduction, estimation, SIR, regression, 3D, plot, graphical representation

See also: MVAppsib, MVAsimdep1, MVAsir2data, MVAsimdepex, MVAppexample, ppsib, ppsibexample

Author: Zografia Anastasiadou

Submitted: Fri, August 05 2011 by Awdesch Melzer

Example:
- 1: The left plots show the response versus the estimated effective dimension reduction directions (EDR-directions). The upper right plot is a three-dimensional plot of the first two directions and the response. The lower right plot shows the eigenvalues (*) and the cumulative sum (o).
- 2: Plot of the true response versus the first true index. The monotonic and the convex shapes can be clearly seen.
- 3: Plot of the true response versus the second true index. The monotonic and the convex shapes can be clearly seen.

B Technologies used

- Charts: draw.io, R
- Programming: R
- Editing: LaTeX
- Project organization: GitHub

C Web links

- Quantnet: http://quantlet.de
- Beta version of Quantnet: http://quantlet.de/d3/beta/
- GitHub Quantnet mirror: https://github.com/QuantLet
- Styleguide for meta-information:
  https://github.com/QuantLet/Styleguide-and-FAQ/blob/master/Styleguide.md
- D3: https://d3js.org
- C3: http://c3js.org
- Examples of QLs for clustering:
  https://github.com/QuantLet/MVA-Ready/tree/master/QID-2548-MVAQnetClusKmeans
  https://github.com/QuantLet/MVA-Ready/tree/master/QID-2544-MVAQnetClusKmeansB
  https://github.com/QuantLet/MVA-Ready/tree/master/QID-2545-MVAQnetClusKmeansT
- Sources for the research: https://github.com/b2net/Clustering_Validation_Pipeline

Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, June 29, 2016

Anastasia Stepanchenko